Chapter 12

Rayleigh quotient minimization

In this chapter we restrict ourselves to the symmetric/Hermitian eigenvalue problem

$$A\mathbf{x} = \lambda M\mathbf{x}, \qquad A = A^*, \quad M = M^* > 0. \tag{12.1}$$

We want to exploit the property of the Rayleigh quotient that

$$\lambda_1 = \min_{\mathbf{x}\neq\mathbf{0}} \rho(\mathbf{x}) \qquad \rho(\mathbf{x}) = \frac{\mathbf{x}^* A\mathbf{x}}{\mathbf{x}^* M\mathbf{x}}, \tag{12.2}$$

which was proved in Theorem 2.15. The basic idea of Rayleigh quotient minimization is to construct a sequence $\{\mathbf{x}_k\}_{k=1,2,\ldots}$ such that $\rho(\mathbf{x}_{k+1}) < \rho(\mathbf{x}_k)$ for all k. The hope is that the sequence $\{\rho(\mathbf{x}_k)\}$ converges to λ_1 and by consequence the vector sequence $\{\mathbf{x}_k\}$ towards the corresponding eigenvector.

The procedure is as follows: For any given $\mathbf{x}_k$ let us choose a **search direction** $\mathbf{p}_k$, so that

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k \mathbf{p}_k. \tag{12.3}$$

The parameter δ_k is determined such that the Rayleigh quotient of the new iterate $\mathbf{x}_{k+1}$ becomes minimal,

$$\rho(\mathbf{x}_{k+1}) = \min_{\delta} \rho(\mathbf{x}_k + \delta\mathbf{p}_k). \tag{12.4}$$

We can write the Rayleigh quotient of the linear combination $\mathbf{x}_k + \delta\mathbf{p}_k$ of two (linearly independant) vectors $\mathbf{x}_k$ and $\mathbf{p}_k$ as

(12.5)

$$\rho(\mathbf{x}_k + \delta\mathbf{p}_k) = \frac{\mathbf{x}_k^* A\mathbf{x}_k + 2\delta\mathbf{x}_k A\mathbf{p}_k + \delta^2 \mathbf{p}_k^* A\mathbf{p}_k}{\mathbf{x}_k^* M\mathbf{x}_k + 2\delta\mathbf{x}_k M\mathbf{p}_k + \delta^2 \mathbf{p}_k^* M\mathbf{p}_k} = \frac{\begin{pmatrix}1\\ \delta\end{pmatrix}^* \begin{bmatrix}\mathbf{x}_k^* A\mathbf{x}_k & \mathbf{x}_k^* A\mathbf{p}_k \\ \mathbf{p}_k^* A\mathbf{x}_k & \mathbf{p}_k^* A\mathbf{p}_k\end{bmatrix} \begin{pmatrix}1\\ \delta\end{pmatrix}}{\begin{pmatrix}1\\ \delta\end{pmatrix}^* \begin{bmatrix}\mathbf{x}_k^* M\mathbf{x}_k & \mathbf{x}_k^* M\mathbf{p}_k \\ \mathbf{p}_k^* M\mathbf{x}_k & \mathbf{p}_k^* M\mathbf{p}_k\end{bmatrix} \begin{pmatrix}1\\ \delta\end{pmatrix}}.$$

This is the Rayleigh quotient associated with the generalized 2×2 eigenvalue problem

$$\begin{bmatrix}\mathbf{x}_k^* A\mathbf{x}_k & \mathbf{x}_k^* A\mathbf{p}_k \\ \mathbf{p}_k^* A\mathbf{x}_k & \mathbf{p}_k^* A\mathbf{p}_k\end{bmatrix} \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \lambda \begin{bmatrix}\mathbf{x}_k^* M\mathbf{x}_k & \mathbf{x}_k^* M\mathbf{p}_k \\ \mathbf{p}_k^* M\mathbf{x}_k & \mathbf{p}_k^* M\mathbf{p}_k\end{bmatrix} \begin{pmatrix}\alpha\\ \beta\end{pmatrix}. \tag{12.6}$$

The smaller of the two eigenvalues of (12.6) is the searched value $\rho_{k+1} := \rho(\mathbf{x}_{k+1})$ in (12.4) that minimizes the Rayleigh quotient. The corresponding eigenvector is normalized such

that its first component equals one[1]. The second component of this eigenvector is $\delta = \delta_k$. Inserting the solution $[1, \delta_k]^*$ into the second line of (12.6) we obtain

$$\mathbf{p}_k^*(A - \rho_{k+1}M)(\mathbf{x}_k + \delta_k \mathbf{p}_k) = \mathbf{p}_k^* \mathbf{r}_{k+1} = 0. \tag{12.7}$$

So, the 'next' residual $\mathbf{r}_{k+1}$ is orthogonal to the actual search direction $\mathbf{p}_k$.

There are various ways how to choose the search direction $\mathbf{p}_k$. A simple way is to cycle through the coordinate vectors, a method that is called coordinate relaxation [3]. It cannot compete with the methods we discuss next.

12.1 The method of steepest descent

Let us make a detour to solving systems of equations

$$A\mathbf{x} = \mathbf{b}, \tag{12.8}$$

where A is symmetric/Hermitian positive definite. Let us define the functional

$$\varphi(\mathbf{x}) \equiv \frac{1}{2}\mathbf{x}^* A\mathbf{x} - \mathbf{x}^*\mathbf{b} + \frac{1}{2}\mathbf{b}^* A^{-1}\mathbf{b} = \frac{1}{2}(A\mathbf{x} - \mathbf{b})^* A^{-1}(A\mathbf{x} - \mathbf{b}). \tag{12.9}$$

The functional φ is minimized (actually zero) at the solution $\mathbf{x}_*$ of (12.8). The negative gradient of φ is

$$-\nabla\varphi(\mathbf{x}) = \mathbf{b} - A\mathbf{x} =: \mathbf{r}(\mathbf{x}). \tag{12.10}$$

It is nonzero except at $\mathbf{x}_*$. In the method of steepest descent [2, 3] a sequence of vectors $\{\mathbf{x}_k\}_{k=1,2,\ldots}$ is constructed such that the relation

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k \mathbf{p}_k. \tag{12.3}$$

holds among any two consecutive vectors. The search direction $\mathbf{p}_k$ is chosen to be the negative gradient $-\nabla\phi(\mathbf{x}_k) = \mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$. This is the direction in which φ decreases the most. Setting $\mathbf{x}_{k+1}$ as in (12.3) we get

$$0 = \left.\frac{\partial\varphi(\mathbf{x}_{k+1})}{\partial\delta}\right|_{\delta=\delta_k} = \mathbf{p}_k^*(A\mathbf{x}_k - \mathbf{b}) + \delta_k \mathbf{p}_k^* A\mathbf{p}_k = -\mathbf{p}_k^*\mathbf{r}_k + \delta_k \mathbf{p}_k^* A\mathbf{p}_k.$$

Thus,

$$\delta_k = \frac{\mathbf{p}_k^*\mathbf{r}_k}{\mathbf{p}_k^* A\mathbf{p}_k} \tag{12.11}$$

which, for steepest descent, becomes

$$\delta_k = \frac{\mathbf{r}_k^*\mathbf{r}_k}{\mathbf{r}_k^* A\mathbf{r}_k} \tag{12.12}$$

Remark 12.1. Notice that

$$\mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1} = \mathbf{b} - A(\mathbf{x}_k + \delta_k \mathbf{p}_k) = \mathbf{r}_k - \delta_k A\mathbf{p}_k. \tag{12.13}$$

[1] The first component of this eigenvector is nonzero if it has a component in the direction of the 'smallest eigenvector'.

Therefore, from (12.11) we have

$$\mathbf{p}_k^* \mathbf{r}_{k+1} = \mathbf{p}_k^* \mathbf{r}_k - \delta_k \mathbf{p}_k^* A \mathbf{p}_k = 0, \tag{12.14}$$

which corresponds to (12.7) in the linear system case. □

For the eigenvalue problem we can proceed similarly by choosing $\mathbf{p}_k$ to be the negative gradient of the Rayleigh quotient ρ,

$$\mathbf{p}_k = -\mathbf{g}_k = -\nabla\rho(\mathbf{x}_k) = -\frac{2}{\mathbf{x}_k^* M \mathbf{x}_k}(A\mathbf{x}_k - \rho(\mathbf{x}_k) M \mathbf{x}_k).$$

Notice that $\mathbf{g}_k$ points in the *same* direction as the residual $\mathbf{r}_k$. (This is in contrast to the linear system case!) Since in eigenvalue problems we only care about directions we can equivalently set

$$\mathbf{p}_k = \mathbf{r}_k = A\mathbf{x}_k - \rho_k M \mathbf{x}_k, \qquad \rho_k = \rho(\mathbf{x}_k). \tag{12.15}$$

With this choice of search direction we immediately have from (12.7) that

$$\mathbf{r}_k^* \mathbf{r}_{k+1} = 0. \tag{12.16}$$

Not surprisingly, the method of steepest descent often converges slowly, as it does for linear systems. This happens if the spectrum is very much spread out, i.e., if the condition number of A relative to B is big.

12.2 The conjugate gradient algorithm

As with linear systems of equations a remedy against the slow convergence of steepest descent are conjugate search directions. So, let's first look at linear systems [5]. There, we define the search directions as[2]

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}, \qquad k > 0. \tag{12.17}$$

where the coefficient β_k is determined such that $\mathbf{p}_k$ and $\mathbf{p}_{k-1}$ are **conjugate**, i.e.,

$$\mathbf{p}_k^* A \mathbf{p}_{k-1} = -\mathbf{g}_k^* A \mathbf{p}_{k-1} + \beta_k \mathbf{p}_{k-1}^* A \mathbf{p}_{k-1} = 0, \tag{12.18}$$

such that

$$\beta_k = \frac{\mathbf{g}_k^* A \mathbf{p}_{k-1}}{\mathbf{p}_{k-1}^* A \mathbf{p}_{k-1}}. \tag{12.19}$$

Premultiplying (12.17) by $\mathbf{g}_k^*$ gives

$$\mathbf{g}_k^* \mathbf{p}_k = -\mathbf{g}_k^* \mathbf{g}_k + \beta_k \mathbf{g}_k^* \mathbf{p}_{k-1} \overset{(12.14)}{=} -\mathbf{g}_k^* \mathbf{g}_k. \tag{12.20}$$

Furthermore,

$$\begin{aligned} 0 \overset{(12.14)}{=} \mathbf{g}_{k+1}^* \mathbf{p}_k &\overset{(12.17)}{=} -\mathbf{g}_{k+1}^* \mathbf{g}_k + \beta_k \mathbf{g}_{k+1}^* \mathbf{p}_{k-1} \\ &\overset{(12.13)}{=} -\mathbf{g}_{k+1}^* \mathbf{g}_k + \beta_k \mathbf{g}_k^* \mathbf{p}_{k-1} + \beta_k \delta_k \mathbf{p}_k^* A \mathbf{p}_{k-1}. \end{aligned}$$

[2] In linear systems the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ is defined as the negative gradient whereas in eigenvalue computations it is defined as $\mathbf{r} = A\mathbf{x} - \rho(\mathbf{x})M\mathbf{x}$, i.e., in the same direction as the gradient. To reduce the confusion we proceed using the gradient.

From (12.14) we have that $\mathbf{g}_k^* \mathbf{p}_{k-1} = 0$ and by construction of $\mathbf{p}_k$ and $\mathbf{p}_{k-1}$ being conjugate we have that $\mathbf{p}_k^* A \mathbf{p}_{k-1} = 0$. Thus,

$$\mathbf{g}_{k+1}^* \mathbf{g}_k = 0, \tag{12.21}$$

as with the method of steepest descent. Still in the case of linear systems, using these identities we find formulae equivalent to (12.19),

$$\beta_k = -\frac{\mathbf{g}_k^* A \mathbf{p}_{k-1}}{\mathbf{p}_{k-1}^* A \mathbf{p}_{k-1}} \overset{(12.13)}{=} -\frac{\mathbf{g}_k^* (\mathbf{g}_k - \mathbf{g}_{k-1})}{\mathbf{p}_{k-1}^* (\mathbf{g}_k - \mathbf{g}_{k-1})} \overset{(12.14)}{=} -\frac{\mathbf{g}_k^* (\mathbf{g}_k - \mathbf{g}_{k-1})}{-\mathbf{p}_{k-1}^* \mathbf{g}_{k-1}}$$

$$\overset{(12.20)}{=} \frac{\mathbf{g}_k^* (\mathbf{g}_k - \mathbf{g}_{k-1})}{\mathbf{g}_{k-1}^* \mathbf{g}_{k-1}} \tag{12.22}$$

$$\overset{(12.21)}{=} \frac{\mathbf{g}_k^* \mathbf{g}_k}{\mathbf{g}_{k-1}^* \mathbf{g}_{k-1}}. \tag{12.23}$$

The equivalent identities (12.19), (12.22), and (12.23) can be used to define β_k the most economic being (12.23).

We now look at how a conjugate gradient algorithm for the eigenvalue problem can be devised. The idea is straightforward. The algorithm differs from steepest descent by the choice of the search direction that are kept conjugate, $\mathbf{p}_{k+1}^* A \mathbf{p}_{k-1} = 0$. are equivalent only when solving linear systems.

The crucial difference to linear systems stems from the fact, that the functional that is to be minimized, i.e., the Rayleigh quotient, is not quadratic anymore. The gradient of $\rho(\mathbf{x})$ is

$$\mathbf{g} = \nabla \rho(\mathbf{x}_k) = \frac{2}{\mathbf{x}^* M \mathbf{x}} (A\mathbf{x} - \rho(\mathbf{x}) M \mathbf{x}).$$

So, in particular, the equation (12.14), does not hold:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k \mathbf{p}_k \quad \not\Longrightarrow \quad \mathbf{g}_{k+1} = \mathbf{g}_k + \delta_k A \mathbf{p}_k.$$

Therefore, in the context of nonlinear systems or eigenvalue problems the formals in (12.19), (12.22), and (12.23) that define β_k are not equivalent anymore! Feng and Owen [4] made comparisons with the three formula and found that in the context of eigenvalue problems the last identity (12.23) leads to the best results. So, we opt for this equation and define the search directions according to

$$\begin{cases} \mathbf{p}_0 = -\mathbf{g}_0, & k = 0, \\ \mathbf{p}_k = -\mathbf{g}_k + \dfrac{\mathbf{g}_k^* M \mathbf{g}_k}{\mathbf{g}_{k-1}^* M \mathbf{g}_{k-1}} \mathbf{p}_{k-1}, & k > 0, \end{cases} \tag{12.24}$$

where we have given the formulae for the generalized eigenvalue problem $A\mathbf{x} = \lambda M \mathbf{x}$. The complete procedure is given in Algorithm 12.1

Convergence

The construction of Algorithm 12.1 guarantees that $\rho(\mathbf{x}_{k+1}) < \rho(\mathbf{x}_k)$ unless $\mathbf{r}_k = \mathbf{0}$, in which case $\mathbf{x}_k$ is the searched eigenvector. In general, i.e., if the initial vector $\mathbf{x}_0$ has a nonvanishing component in the direction of the 'smallest' eigenvector $\mathbf{u}_1$, convergence is toward the smallest eigenvalue λ_1. This assumption must also hold for vector iteration or the Lanczos algorithm.

Algorithm 12.1 The Rayleigh-quotient algorithm

Let $\mathbf{x}_0$ be a unit vector, $\|\mathbf{x}_0\|_M = 1$.
$\mathbf{v}_0 := A\mathbf{x}_0, \quad \mathbf{u}_0 := M\mathbf{x}_0,$
$\rho_0 := \dfrac{\mathbf{v}_0^*\mathbf{x}_0}{\mathbf{u}_0^*\mathbf{x}_0},$
$\mathbf{g}_0 := 2(\mathbf{v}_0 - \rho_0\mathbf{u}_0)$
while $\|\mathbf{g}_k\| > tol$ **do**
  if $k = 1$ **then**
    $\mathbf{p}_k := -\mathbf{g}_{k-1}$;
  else
    $\mathbf{p}_k := -\mathbf{g}_{k-1} + \dfrac{\mathbf{g}_{k-1}^* M \mathbf{g}_{k-1}}{\mathbf{g}_{k-2}^* M \mathbf{g}_{k-2}} \mathbf{p}_{k-1}$;
  end if
  Determine the smallest Ritz value and corresponding Ritz vector $\mathbf{x}_k$ of (A, M) in $\mathcal{R}([\mathbf{x}_{k-1}, \mathbf{p}_k])$
  $\mathbf{v}_k := A\mathbf{x}_k, \quad \mathbf{u}_k := M\mathbf{x}_k$
  $\rho_k := \mathbf{x}_k^*\mathbf{v}_k / \mathbf{x}_k^*\mathbf{u}_k$
  $\mathbf{g}_k := 2(\mathbf{v}_k - \rho_k\mathbf{u}_k)$
end while

Let

$$\mathbf{x}_k = \cos\vartheta_k \mathbf{u}_1 + \sin\vartheta_k \mathbf{z}_k =: \cos\vartheta_k \mathbf{u}_1 + \mathbf{w}_k, \tag{12.25}$$

where $\|\mathbf{x}_k\|_M = \|\mathbf{u}_1\|_M = \|\mathbf{z}_k\|_M = 1$ and $\mathbf{u}_1^* M \mathbf{z}_k = 0$. Then we have

$$\begin{aligned}\rho(\mathbf{x}_k) &= \cos^2\vartheta_k \lambda_1 + 2\cos\vartheta_k \sin\vartheta_k \mathbf{u}_1^* A \mathbf{z}_k + \sin^2\vartheta_k \mathbf{z}_k^* A \mathbf{z}_k \\ &= \lambda_1(1 - \sin^2\vartheta_k) + \sin^2\vartheta_k \rho(\mathbf{z}_k),\end{aligned}$$

or,

$$\rho(\mathbf{x}_k) - \lambda_1 = \sin^2\vartheta_k \,(\rho(\mathbf{z}_k) - \lambda_1) \geq (\lambda_2 - \lambda_1)\sin^2\vartheta_k. \tag{12.26}$$

As seen earlier, in symmetric eigenvalue problems, the eigenvalues are much more accurate than the eigenvectors.

Let us now suppose that the eigenvectors have already converged, i.e.,

$$\rho(\mathbf{x}_k) = \rho_k \cong \lambda_1,$$

while the eigenvectors are not yet as accurate as desired. Then we can write

$$\mathbf{r}_k = (A - \rho_k M)\mathbf{x}_k \cong (A - \lambda_1 M)\mathbf{x}_k = \sum_{j=1}^{n} (\lambda_j - \lambda_1) M \mathbf{u}_j \mathbf{u}_j^* M \mathbf{x}_k \tag{12.27}$$

which entails $\mathbf{u}_1^* \mathbf{r}_k = 0$ since the first summand on the right of (12.27) vanishes. From (12.25) we have $\mathbf{w}_k = \sin\vartheta_k \mathbf{z}_k \perp_M \mathbf{u}_1$. Thus,

$$\begin{cases} (A - \lambda_1 M)\mathbf{w}_k = (A - \lambda_1 M)\mathbf{x}_k = \mathbf{r}_k \perp \mathbf{u}_1 \\ \qquad\qquad \mathbf{w}_k^* M \mathbf{u}_1 = 0 \end{cases} \tag{12.28}$$

If λ_1 is a simple eigenvalue of the pencil $(A; B)$ then $A - \lambda_1 M$ is a bijective mapping of $\mathcal{R}(\mathbf{u}_1)^{\perp_M}$ onto $\mathcal{R}(\mathbf{u}_1)^{\perp}$. If $\mathbf{r} \in \mathcal{R}(\mathbf{u}_1)^{\perp}$ then the equation

$$(A - \lambda_1 M)\mathbf{w} = \mathbf{r} \tag{12.29}$$

has a *unique* solution in $\mathcal{R}(\mathbf{u}_1)^{\perp_M}$.

So, close to convergence, Rayleigh–Quotient minimization does nothing else but solving equation (12.29). Since the solution is in the Krylov subspace $\mathcal{K}^j\left((A-\lambda_1 M)\mathbf{g}_j\right)$ for some j, the orthogonality condition $\mathbf{w}_k^* M\mathbf{u}_1$ is implicitly fulfilled. The convergence of the Rayleigh–Quotient minimization is determined by the condition number of $A-\lambda_1 M$ (as a mapping of $\mathcal{R}(\mathbf{u}_1)^{\perp_M}$ onto $\mathcal{R}(\mathbf{u}_1)^{\perp}$), according to the theory of conjugate gradients for linear system of equations. This condition number is

$$\kappa_0 = \mathcal{K}(A-\lambda_1 M)\Big|_{\mathcal{R}(\mathbf{u}_1)^{\perp_M}} = \frac{\lambda_n-\lambda_1}{\lambda_2-\lambda_1}, \tag{12.30}$$

and the rate of convergence is given by

$$\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}. \tag{12.31}$$

A high condition number implies slow convergence. We see from (12.31) that the condition number is high if the distance of λ_1 and λ_2 is much smaller than the spread of the spectrum of $(A;B)$. This happens more often than not, in particular with FE discretizations of PDE's.

Preconditioning

In order to reduce the condition number of the eigenvalue problem we change

$$A\mathbf{x} = \lambda M\mathbf{x}$$

in

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\lambda}\tilde{M}\tilde{\mathbf{x}} \tag{12.32}$$

such that

$$\kappa(\tilde{A}-\tilde{\lambda}_1\tilde{M}) \ll \kappa(A-\lambda_1 M). \tag{12.33}$$

To further investigate this idea, let C be a nonsingular matrix, and let $\mathbf{y}=C\mathbf{x}$. Then,

$$\rho(\mathbf{x}) = \frac{\mathbf{x}^*A\mathbf{x}}{\mathbf{x}^*M\mathbf{x}} = \frac{\mathbf{y}^*C^*AC^{-1}\mathbf{y}}{\mathbf{y}^*C^*MC^{-1}\mathbf{y}} = \frac{\mathbf{y}^*\tilde{A}\mathbf{y}}{\tilde{\mathbf{y}}^*\tilde{M}\mathbf{y}} = \tilde{\rho}(\mathbf{y}) \tag{12.34}$$

Thus,

$$\tilde{A}-\lambda_1\tilde{M} = C^{-*}(A-\lambda_1 M)C^{-1},$$

or, after a similarity transformation,

$$C^{-1}(\tilde{A}-\lambda_1\tilde{M})C = (C^*C)^{-1}(A-\lambda_1 M).$$

How should we choose C to satisfy (12.33)? Let us tentatively set $C^*C=A$. Then we have

$$(C^*C)^{-1}(A-\lambda_1 M)\mathbf{u}_j = A^{-1}(A-\lambda_1 M)\mathbf{u}_j = (I-\lambda_1 A^{-1}M)\mathbf{u}_j = \left(1-\frac{\lambda_1}{\lambda_j}\right)\mathbf{u}_j.$$

Note that

$$0 \le 1-\frac{\lambda_1}{\lambda_j} < 1.$$

Dividing the largest eigenvalue of $A^{-1}(A-\lambda_1 M)$ by the smallest *positive* gives the condition number

$$\kappa_1 := \kappa\left(A^{-1}(A-\lambda_1 M)\big|_{\mathcal{R}(\mathbf{u}_1)^{\perp_M}}\right) = \frac{1-\frac{\lambda_1}{\lambda_n}}{1-\frac{\lambda_1}{\lambda_2}} = \frac{\lambda_2}{\lambda_n}\frac{\lambda_n-\lambda_1}{\lambda_2-\lambda_1} = \frac{\lambda_2}{\lambda_n}\kappa_0. \tag{12.35}$$

If $\lambda_2 \ll \lambda_n$ then the condition number is heavily reduced. Further, κ_1 is bounded independently of n,

$$\kappa_1 = \frac{1-\lambda_1/\lambda_n}{1-\lambda_1/\lambda_2} < \frac{1}{1-\lambda_1/\lambda_2}. \tag{12.36}$$

So, with this particular preconditioner, κ_1 does not dependent on the choice of the mesh-width h in the FEM application.

The previous discussion suggests to choose C in such way that $C^*C \cong A$. C could, for instance, be obtained form an Incomplete Cholesky decomposition. We make this choice in the numerical example below.

Notice that the transformation $\mathbf{x} \longrightarrow \mathbf{y} = C\mathbf{x}$ need not be done explicitly. In particular the matrices $\tilde{A}$ and $\tilde{M}$ must not be formed. As with the preconditioned conjugate gradient algorithm for linear systems there is an additional step in the algorithm where the *preconditioned residual* is computed, see Fig. 12.1.

12.3 Locally optimal PCG (LOPCG)

The parameter δ_k in the RQMIN und (P)CG algorithms is determined such that

$$\rho(\mathbf{x}_{k+1}) = \rho(\mathbf{x}_k + \delta_k \mathbf{p}_k), \quad \mathbf{p}_k = -\mathbf{g}_k + \alpha_k \mathbf{p}_{k-1} \tag{12.37}$$

is minimized. α_k is chosen to make consecutive search directions conjugate. Knyazev [6] proposed to optimize both parameters, α_k and δ_k, at once.

$$\rho(\mathbf{x}_{k+1}) = \min_{\delta,\gamma} \rho(\mathbf{x}_k - \delta \mathbf{g}_k + \gamma \mathbf{p}_{k-1}) \tag{12.38}$$

This results in potentially smaller values for the Rayleigh quotient, as

$$\min_{\delta,\gamma} \rho\big(\mathbf{x}_k - \delta \mathbf{g}_k + \gamma \mathbf{p}_{k-1}\big) \le \min_{\delta} \big(\mathbf{x}_k - \delta(\mathbf{g}_k - \alpha_k \mathbf{p}_k)\big).$$

Hence, Knyazev coined the notation "locally optimal".

$\rho(\mathbf{x}_{k+1})$ in (12.38) is the minimal eigenvalue of the 3×3 eigenvalue problem

$$\begin{bmatrix} \mathbf{x}_k^* \\ -\mathbf{g}_k^* \\ \mathbf{p}_{k-1}^* \end{bmatrix} A[\mathbf{x}_k, -\mathbf{g}_k, \mathbf{p}_{k-1}] \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \lambda \begin{bmatrix} \mathbf{x}_k^* \\ -\mathbf{g}_k^* \\ \mathbf{p}_{k-1}^* \end{bmatrix} M[\mathbf{x}_k, -\mathbf{g}_k, \mathbf{p}_{k-1}] \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \tag{12.39}$$

We normalize the eigenvector corresponding to the smallest eigenvalue such that its first component becomes 1,

$$[1, \delta_k, \gamma_k] := [1, \beta/\alpha, \gamma/\alpha].$$

These values of δ_k and γ_k are the parameters that minimize the right hand side in (12.38). Then we can write

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \delta_k \mathbf{g}_k + \gamma_k \mathbf{p}_{k-1} = \mathbf{x}_k + \delta_k \underbrace{(-\mathbf{g}_k + (\gamma_k/\delta_k)\mathbf{p}_{k-1})}_{=:\mathbf{p}_k} = \mathbf{x}_k + \delta_k \mathbf{p}_k. \tag{12.40}$$

We can consider $\mathbf{x}_{k+1}$ as having been obtained by a Rayleigh quotient minimization from $\mathbf{x}_k$ along $\mathbf{p}_k = -\mathbf{g}_k + (\gamma_k/\delta_k)\mathbf{p}_{k-1}$. Notice that this direction in needed in the next iteration step. (Otherwise it is not of a particular interest.)

```
function [x,rho,log] = rqmin1(A,M,x,tol,C)
%RQMIN1    [x,rho] = rqmin1(A,M,x0,tol,C)
%       cg-Rayleigh-Quotienten-Minimization for the computation
%       of the smallest eigenvalue of  A*x = lambda*M*x,
%       A and M are symmetric, M spd.  x0 initial vector
%       C'*C preconditioner
%       tol: convergence criterium:
%            ||2*(C'*C)\(A*x - lam*M*x)|| < tol

% PA 16.6.2000

u = M*x;
q = sqrt(x'*u);
x = x/q; u = u/q;
v = A*x;
rho = x'*v;

k = 0; g = x; gnorm = 1;  log=[]; % Initialisierungen

while gnorm > tol,
  k = k + 1;
  galt = g;
  if exist('C'),
    g = 2*(C\(C'\(v - rho*u)));   %  vorkonditionierter Gradient
  else
    g = 2*(v - rho*u);      %  Gradient
  end
  if k == 1,
    p = -g;
  else
    p = -g + (g'*M*g)/(galt'*M*galt)*p;
  end

  [qq,ll] = eig([x p]'*[v A*p],[x p]'*[u M*p]);
  [rho,ii] = min(diag(ll));
  delta = qq(2,ii)/qq(1,ii);

  x = x + delta*p;
  u = M*x;
  q = sqrt(x'*u);
  x = x/q; u = u/q;
  v = A*x;
  gnorm = norm(g);
  if nargout>2, log = [log; [k,rho,gnorm]]; end
end
```

Figure 12.1: MATLAB code RQMIN: Rayleigh quotient minimization

```
function [x,rho,log] = lopcg(A,M,x,tol,C)
%RQMIN1 [x,rho] = lopcg(A,M,x0,tol,C)
%       Locally Optimal Proconditioned CG algorithm for
%       computing the smallest eigenvalue of  A*x = lambda*M*x,f
%       where A and M are symmetrisch, M spd.
%       x0 initial vektor
%       C'*C preconditioner
%       tol: stopping criterion:
%            (C'*C)\(A*x - lam*M*x) < tol

% PA 2002-07-3

n = size(M,1);
u = M*x;
q = sqrt(x'*u);
x = x/q; u = u/q;
v = A*x;
rho = x'*v;

k = 0;  gnorm = 1;  log=[]; % initializations

while gnorm > tol,
  k = k + 1;
  g = v - rho*u;        %  gradient
  gnorm = norm(g);
  if exist('C'),
    g = (C\(C'\g));   %  preconditioned gradient
  end
  if k == 1, p = zeros(n,0); end

  aa = [x -g p]'*[v A*[-g p]]; aa = (aa+aa')/2;
  mm = [x -g p]'*[u M*[-g p]]; mm = (mm+mm')/2;
  [qq,ll] = eig(aa,mm);
  [rho,ii] = min(diag(ll));
  delta = qq(:,ii);

  p = [-g p]*delta(2:end);
  x = delta(1)*x + p;
  u = M*x;
  q = sqrt(x'*u);
  x = x/q; u = u/q;
  v = A*x;
  if nargout>2, log = [log; [k,rho,gnorm]]; end
end
```

Figure 12.2: MATLAB code LOPCG: Locally Optimal Preconditioned Conjugate Gradient algorithm

12.4 The block Rayleigh quotient minimization algorithm (BRQMIN)

The above procedures converge *very* slowly if the eigenvalues are clustered. Hence, these methods should be applied only in **blocked** form.

Longsine and McCormick [7] suggested several variants for blocking Algorithm 12.1. See [1] for a recent numerical investigation of this algorithm.

12.5 The locally-optimal block preconditioned conjugate gradient method (LOBPCG)

In BRQMIN the Rayleigh quotient is minimized in the $2q$-dimensional subspace generated by the eigenvector approximations X_k and the search directions $P_k = -H_k + P_{k-1}B_k$, where the H_k are the preconditioned residuals corresponding to X_k and B_k is chosen such that the *block* of search directions is conjugate. Instead, Knyazev [6] suggests that the space for the minimization be augmented by the q-dimensional subspace $\mathcal{R}(H_k)$. The resulting algorithm is deemed 'locally-optimal' because $\rho(\mathbf{x})$ is minimized with respect to all available vectors.

Algorithm 12.2 The locally-optimal block preconditioned conjugate gradient method (LOBPCG) for solving $A\mathbf{x} = \lambda M\mathbf{x}$ with preconditioner N of [1]

Choose random matrix $X_0 \in \mathbb{R}^{n\times q}$ with $X_0^T M X_0 = I_q$. Set $Q := [\,]$.
Compute $(X_0^T K X_0)S_0 = S_0\Theta_0$ /* (Spectral decomposition) */
where $S_0^T S_0 = I_q$, $\Theta_0 = \mathrm{diag}(\vartheta_1, \ldots, \vartheta_q)$, $\vartheta_1 \le \ldots \le \vartheta_q$.
$X_0 := X_0 S_0$; $R_0 := KX_0 - MX_0\Theta_0$; $P_0 := [\,]$; $k := 0$.
while $\mathrm{rank}(Q) < p$ **do**
  Solve the preconditioned linear system $NH_k = R_k$
  $H_k := H_k - Q(Q^T M H_k)$.
  $\widetilde{K} := [X_k, H_k, P_k]^T K [X_k, H_k, P_k]$.
  $\widetilde{M} := [X_k, H_k, P_k]^T M [X_k, H_k, P_k]$.
  Compute $\widetilde{K}\widetilde{S}_k = \widetilde{M}\widetilde{S}_k\widetilde{\Theta}_k$ /* (Spectral decomposition) */
  where $\widetilde{S}_k^T\widetilde{M}\widetilde{S}_k = I_{3q}$, $\widetilde{\Theta}_k = \mathrm{diag}(\vartheta_1, \ldots, \vartheta_{3q})$, $\vartheta_1 \le \ldots \le \vartheta_{3q}$.
  $S_k := \widetilde{S}_k[\mathbf{e}_1, \ldots, \mathbf{e}_q]$, $\Theta := \mathrm{diag}(\vartheta_1, \ldots, \vartheta_q)$.
  $P_{k+1} := [H_k, P_k]\, S_{k,2}$; $X_{k+1} := X_k S_{k,1} + P_{k+1}$.
  $R_{k+1} := KX_{k+1} - MX_{k+1}\Theta_k$.
  $k := k + 1$.
  for $i = 1, \ldots, q$ **do**
    /* (Convergence test) */
    if $\|R_k\mathbf{e}_i\| <$ tol **then**
      $Q := [Q, X_k\mathbf{e}_i]$; $X_k\mathbf{e}_i := \mathbf{t}$, with $\mathbf{t}$ a random vector.
      M-orthonormalize the columns of X_k.
    end if
  end for
end while

If $\mathbf{d}_j = [\mathbf{d}_{1j}^T, \mathbf{d}_{2j}^T, \mathbf{d}_{3j}^T]^T$, $\mathbf{d}_{ij} \in \mathbb{R}^q$, is the eigenvector corresponding to the j-th eigenvalue of (12.1) restricted to $\mathcal{R}([X_k, H_k, P_{k-1}])$, then the j-th column of X_{k+1} is the corresponding

Ritz vector

$$X_{k+1}\mathbf{e}_j := [X_k, H_k, P_{k-1}]\ \mathbf{d}_j = X_k\mathbf{d}_{1j} + P_k\mathbf{e}_j, \tag{12.41}$$

with

$$P_k\mathbf{e}_j := H_k\mathbf{d}_{2j} + P_{k-1}\mathbf{d}_{3j}.$$

Notice that P_0 is an empty matrix such that the eigenvalue problem in step (8) of the locally-optimal block preconditioned conjugate gradient method (LOBPCG), displayed in Algorithm 12.2, has order $2q$ only for $k = 0$.

The algorithm as proposed by Knyazev [6] was designed to compute just a few eigenpairs and so a memory efficient implementation was not presented. For instance, in addition to X_k, R_k, H_k, P_k, the matrices MX_k, MH_k, MP_k and KX_k, KH_k, KP_k are also stored. The resulting storage needed is prohibitive if more than a handful of eigenpairs are needed.

A more memory efficient implementation results when we iterate with blocks of width q in the space orthogonal to the already computed eigenvectors. The computed eigenvectors are stored in Q and neither MQ nor KQ are stored. Hence only storage for $(p + 10q)n + \mathcal{O}(q^2)$ numbers is needed.

Here, the columns of $[X_k, H_k, P_k]$ may become (almost) linearly dependent leading to ill-conditioned matrices $\widetilde{K}$ and $\widetilde{M}$ in step (9) of the LOBPCG algorithm. If this is the case we simply restart the iteration with random X_k orthogonal to the computed eigenvector approximations. More sophisticated restarting procedures that retain X_k but modify H_k and/or P_k were much less stable in the sense that the search space basis again became linearly dependent within a few iterations. Restarting with random X_k is a rare occurrence and in our experience, has little effect on the overall performance of the algorithm.

12.6 A numerical example

We again look at the determination the acoustic eigenfrequencies and modes in the interior of a car, see section 1.6.3. The computations are done with the finest grid depicted in Fig. 1.9. We compute the smallest eigenvalue of the problem with RQMIN and LOPCG, with preconditioning and without. The preconditioner we chose was the incomplete Cholesky factorization without fill-in, usually denoted IC(0). This factorization is implemented in the MATLAB routine `cholinc`.

```
>> [p,e,t]=initmesh('auto');
>> [p,e,t]=refinemesh('auto',p,e,t);
>> [p,e,t]=refinemesh('auto',p,e,t);
>> p=jigglemesh(p,e,t);
>> [A,M]=assema(p,t,1,1,0);
>> whos
  Name      Size                    Bytes  Class

  A      1095x1095                  91540  double array (sparse)
  M      1095x1095                  91780  double array (sparse)
  e         7x188                   10528  double array
  p         2x1095                  17520  double array
  t         4x2000                  64000  double array

Grand total is 26052 elements using 275368 bytes
```

```
>> n=size(A,1);
>> R=cholinc(A,'0');   % Incomplete Cholesky factorization
>> x0=rand(n,1)-.5;
>> [x,rho,log0] = rqmin1(A,M,x0,tol);
>> [x,rho,log1] = rqmin1(A,M,x0,tol,R);
>> [x,rho,log2] = lopcg(A,M,x0,tol);
>> [x,rho,log3] = lopcg(A,M,x0,tol,R);
>> whos log*
  Name       Size                    Bytes  Class

  log0     346x3                      8304  double array
  log1     114x3                      2736  double array
  log2     879x3                     21096  double array
  log3     111x3                      2664  double array

Grand total is 4350 elements using 34800 bytes

>> L = sort(eig(full(A),full(M)));
>> format short e, [L(1) L(2) L(n)], format

ans =

  -7.5901e-13   1.2690e-02   2.6223e+02

>> k0= L(n)/L(2);
>> (sqrt(k0) - 1)/(sqrt(k0) + 1)

ans =

    0.9862

>> l0=log0(end-6:end-1,2).\log0(end-5:end,2);
>> l1=log1(end-6:end-1,2).\log1(end-5:end,2);
>> l2=log2(end-6:end-1,2).\log2(end-5:end,2);
>> l3=log3(end-6:end-1,2).\log3(end-5:end,2);
>> [l0 l1 l2 l3]

ans =

    0.9292    0.8271    0.9833    0.8046
    0.9302    0.7515    0.9833    0.7140
    0.9314    0.7902    0.9837    0.7146
    0.9323    0.7960    0.9845    0.7867
    0.9320    0.8155    0.9845    0.8101
    0.9301    0.7955    0.9852    0.8508

>> semilogy(log0(:,1),log0(:,3)/log0(1,3),log1(:,1),log1(:,3)/log1(1,3),...
   log2(:,1),log2(:,3)/log2(1,3),log3(:,1),log3(:,3)/log3(1,3),'LineWidth',2)
>> legend('rqmin','rqmin + prec','lopcg','lopcg + prec')
```

The convergence histories in Figure 12.3 for RQMIN and LOPCG show that preconditioning helps very much in reducing the iteration count.

In Figure 12.4 the convergence histories of LOBPCG for computing ten eigenvalues is shown. In 43 iteration steps all *ten* eigenvalues have converged to the desired accuracy ($\varepsilon = 10^{-5}$). Clearly, the iteration count has been decreased drastically. Note however,

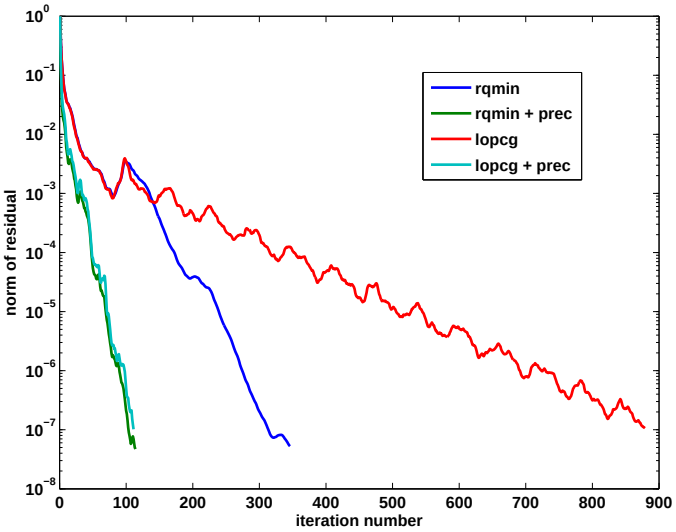

Figure 12.3: Convergence of variants of Rayleigh quotient minimization

that each iteration step requires solving ten systems of equation resulting in 430 system solves. (In fact, if converged eigenvectors are locked, only 283 systems had to be solved.) Nevertheless, when comparing with Fig. 12.3 one should remember that in the LOBPCG computation ten eigenpairs have been computed. If a single eigenpair is required then a blocksize of 10 is too big, but a smaller blocksize may reduce the execution time. If a small number of eigenvalues is desired then a blocksize equal or slightly bigger than theis number is certainly advantageous. Not that in step (5) of Algorithm 12.2 q linear systems of equations are solved concurrently. An efficient implementation accesses the preconditioner N only once. The MATLAB code does this naturally.

Bibliography

[1] P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. Tuminaro, *A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 204–236.

[2] O. Axelsson and V. Barker, *Finite Element Solution of Boundary Value Problems*, Academic Press, Orlando FL, 1984.

[3] D. K. Faddeev and V. N. Faddeeva, *Computational Methods of Linear Algebra*, Freeman, San Francisco, 1963.

[4] Y. T. Feng and D. R. J. Owen, *Conjugate gradient methods for solving the smallest eigenpair of large symmetric eigenvalue problems*, Internat. J. Numer. Methods Engrg., 39 (1996), pp. 2209–2229.

[5] M. R. Hestenes and E. Stiefel, *Methods of conjugent gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[6] A. V. Knyazev, *Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.

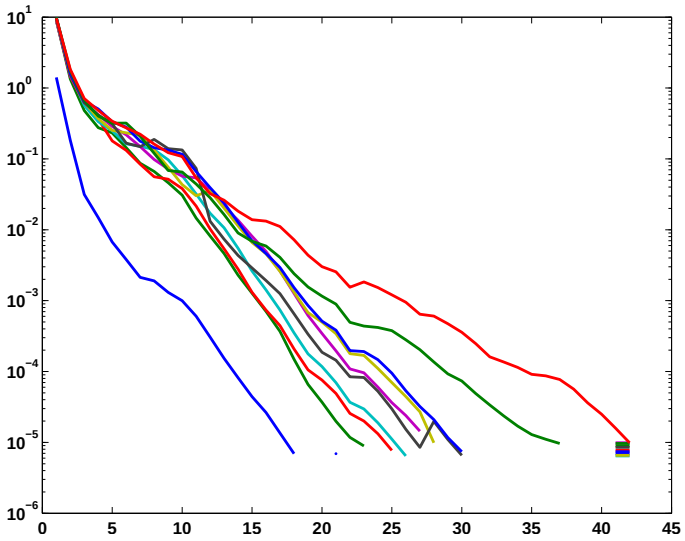

Figure 12.4: Convergence of 10 eigenvalues with LOBPCG preconditioned by IC(0)

[7] D. E. Longsine and S. F. McCormick, *Simultaneous Rayleigh–quotient minimization methods for* $Ax = \lambda Bx$, Linear Algebra Appl., 34 (1980), pp. 195–234.

[8] A. Ruhe, *Computation of eigenvalues and vectors*, in Sparse Matrix Techniques, V. A. Barker, ed., Lecture Notes in Mathematics 572, Berlin, 1977, Springer-Verlag, pp. 130–184.

[9] H. R. Schwarz, *Rayleigh–Quotient–Minimierung mit Vorkonditionierung*, in Numerical Methods of Approximation Theory, Vol. 8, L. Collatz, G. Meinardus, and G. Nürnberger, eds., vol. 81 of International Series of Numerical Mathematics (ISNM), Basel, 1987, Birkhäuser, pp. 229–45.